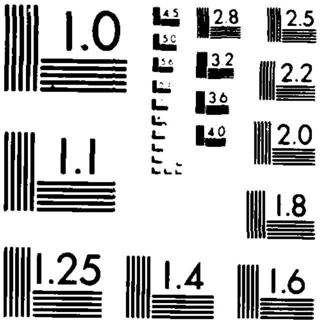
AD-A175 189 THE COUNSELOR PROJECT: UNDERSTANDING ANSWERS TO 1/1
QUESTIONS(U) MASSACHUSETTS UNIV AMHERST DEPT OF
COMPUTER AND INFORMATION SCIENCE W G LEHNERT ET AL.
UNCLASSIFIED FEB 86 CPTM-9 N00014-85-K-0017 F/G 5/7 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

The Counselor Project

Department of Computer and Information Science
University of Massachusetts
Amherst, Mass. 01003

Understanding Answers to Questions

Wendy G. Lehnert and Brian K. Stucky

February, 1986

CPTM #9

This series of internal memos describes research in
artificial intelligence conducted under
DARPA Contract N00014-85-K-0017

2

Understanding Answers to Questions

Wendy G. Lehnert and Brian K. Stucky

February, 1986

CPTM #9

This research was supported by DARPA grant N00014-85-K-0017.

1. INTRODUCTION

Typically, a computational model of question answering addresses the problem of understanding questions and producing answers. Once a question has been understood, the system accesses a knowledge base to retrieve the appropriate answer. Research motivated by the COUNSELOR system, a legal reasoning program, has turned the tables on this view of question answering. In COUNSELOR, questions are posed by the system and the natural language interface must understand the answer. Pursuant to these needs, we present a system model that utilizes the notion of predictive processing in natural language understanding to interpret user answers to system questions.

When we ask questions, we are seeking certain types of information. If we ask a yes/no question, we are hoping to confirm or deny a hypothesis that we have. In the case of Wh-questions, we are looking for new information that we do not know. Regardless of the type of question, we create expectations for the answer we want to receive. A precise and careful presentation of the question can limit possible responses. With a Q/A computational model of the type indicated earlier, the answer must be interpreted in whatever context and form it has been presented. Consequently, a mechanism that understands answers to questions must be robust enough to handle a diverse set of potential constructions. These needs have allowed two previously distinct lines of research to merge. The first is the unique approach to question/answer understanding. The second is the application of predictive processing in natural language understanding systems to this task. Much work has been done on predictive parsing in analyzing both sentences and continuous text (Riesbeck & Schank, 1976; DeJong, 1979; Charniak, 1981; Waltz & Pollack, 1984).

However, these techniques have not been utilized to deal with answers to questions. The needs of a question asking system can very naturally be met with predictive natural language processing techniques.

The COUNSELOR project is composed of several units (Ashley & Rissland, 1985; McDonald & Pustejovsky, 1985; Lehnert & Rosenberg, 1985). PLUM, the natural language processing component, acts as the interface between users and the rest of COUNSELOR. The model we will discuss is based on the operation of this component. It is important to distinguish two common methods of developing natual language understanding systems. (1) Modular parsing separates the components concerned with parsing and memory (Woods, 1970; Bates, 1978; Marcus, 1980). The processing of a parser in this type of system is independent of any memory processes or content. Modular parsing systems typically produce syntactic parse trees as output. (2) Integrated language analysis combines sentence analysis with processes normally associated with memory (Dyer, 1983; Riesbeck & Martin, 1986). This makes it possible to parse sentences in a context; both world and domain. Integrated parsers usually produce an internal meaning representation of the text. This is the approach utilized by COUNSELOR. Initially, we explore some of the features of questions and answers that our model must accommodate.

2. BACKGROUND

We will begin by viewing some examples of the question answering process to identify necessary features for a computational model. This discussion will focus primarily on yes/no questions although, as we shall see later, some of these concepts are applicable to the class of Wh-questions as well.

2.1 CATEGORIES OF Y/N QUESTIONS

Various categories of questions have been suggested for everyday discourse (Lehnert, 1978; Graesser & Black, 1985). We will restrict ourselves to the class of yes/no questions, and attempt to subdivide this general Q/A interaction further. The categorization presented here is based on the form taken by the answer. Features of the question's response, both structural and conceptual, can help to distinguish yes/no response types. Six initial categories are noted:

(1) Simple response.

Did Soleil work on the Autotell project?
Yes.

Was Mary in an accident?
No, she wasn't.

Confirmation or denial of the question is made explicit by means of a keyword or keyword preface (yes/no). A pronominal reference to some constituent of the question may also be present.

(2) Response with gratuitous information.

Do you like their hamburgers?

Yes, and their french fries too.

Is their food good?

No, and their service is poor as well.

As with the prior category, a confirmation or denial is clear from the use of the keyword. In addition, unsolicited information that is not specifically related to the main topic of the question must be present. This additional information is relevant to the conceptual import of the question. For example, we would not expect a question/answer pair such as:

Do you like their hamburgers?

Yes, and the weather too.

(3) Echo response.

Did you enjoy John's speech?

I did not.

Is this just a specialization of Val's work?

This is a specialization of Val's work.

There is no explicit keyword confirmation or denial as we saw in the previous categories. The answer is implicit within the answer ellipsis or question rephrasal. In fact, the answer itself is a declarative statement that can be derived by doing a simple syntactic transformation of the sentence subject and auxiliary. Negation of the verb indicates a denial of the question.

(4) Qualified response.

Is the humid weather bothering you?

Only at night.

Does Howard want to see a movie?

He would like to see a comedy.

These answers are termed *qualified* since they are restricted to some aspect of the question content. In both examples, the answer is "yes," but only within the meaning of the specified query (only at a certain time, or only if the movie is a certain type). This category might be viewed as though several questions were being implicitly asked, each covering a particular restriction. For example:

Is the humid weather bothering you during the day? No.

Is the humid weather bothering you at night? Yes.

Does Howard want to see a funny movie? Yes.

Does Howard want to see a dramatic movie? No.

(5) Inconclusive response.

Do you want to go out to eat?

Not at McDonald´s.

Does Bob like soccer?

He hates most sports.

This type of answer is similar to a qualified response, but the categories may be distinguished. Once again, we have a qualified response to the question. However, unlike the qualified response category, the answer is not clear. In this category a follow-up question is usually necessary to determine the answer. A confirmation or denial of the question is difficult to assess. We might expect the following logical follow-up questions that attempt to elicit a clear response:

Do you want to go out to eat?

Not at McDonald´s.

Would you like to eat somewhere else then?

Does Bob like soccer?

He hates most sports.

But does he hate soccer?

(6) Emphatic response.

Did Soleil work on the Autotell project?

Yes, he was a key employee on the project.

Do you like Mary?

No, she's my worst enemy.

This category is similar to a response with gratuitous information, but a distinction may be drawn on the relevance or specificity of the additional information. Again, the explicit confirmation is followed by unsolicited data. However, in this case, the data emphasizes the already confirmed (or denied) topic of the question. It specifically relates to the question topic, not just the general question content as we saw in the category of gratuitous information. An emphatic answer makes a response more specific than the original question. A response with gratuitous information merely adds restrictions to the original question.

We might view all these categories as falling into two distinct classes: the **prefaced class**, those with a keyword (simple, gratuitous information, and emphatic response) and the **open class**, those without a keyword preface (echo, qualified, and inconclusive response). In the case of the prefaced response, understanding the answer in terms of the question is simple; we need only look at the keyword. The only challenge arises in understanding additional information. The open class presents more problems. Here, interpreting a confirmation or denial is usually more difficult. The one exception is, perhaps, the class of echo responses. But even then, more processing must be done. Understanding in the open class and, to some extent, in the preface class, is dependent upon a determination of the question thrust and context. This notion, one of question focus, becomes crucial as the model is developed.

2.2 QUESTION FOCUS

Focus may be viewed in several different ways. We can analyze the focus of a discourse in terms of its conceptual topic (Grosz, 1978, 1981; Sidner, 1983). Linguists can view a syntactic focus on a strictly grammatical level. We are concerned solely with the focus of questions. This focus is based on the conceptual component of the question to which attention is directed, and it can be useful in understanding answers to Y/N questions. Four distinct types of focus may be examined: stress-intonation patterns, syntactic constructions, context, and world knowledge. We will be primarily concerned with the category of focus involving the actual syntactic construction of the question. The latter two categories involve conceptual processing at a level beyond the scope of this model.

(1) Stress-Intonation patterns.

The focus of the question may be determined by examining the intonation of the speaker's voice as the question is uttered (Davies & Isard, 1972). Accentuating various components of the question can convey to the hearer expectations of the speaker. For example:

Did *John* bring Bill a gift?

The focus in this question, based on intonation, is John. Is John the one who brought Bill a gift?

Did John bring *Bill* a gift?

Here the focus is on Bill. Was it Bill who received a gift from John?

Did John bring Bill a *gift*?

Finally, we have the focus placed on the gift. Was it a gift that John brought Bill?

Stress intonation patterns are only useful for spoken language and, therefore, not relevant, for computational models of written questions.

(2) Syntactic contructions.

The actual structure of the question can provide clues to its focus. In fact, the focus resulting from syntactic constructions will allow us to create expectations for a given type of answer. For example:

Q1: Was it the waitress who brought John a menu?

Here the construction of the question places focus on the waitress. Therefore, we would expect answers such as:

A1: No, the hostess brought John a menu.

Because of the syntactic structure, we would not anticipate an answer like:

A2: No, the waitress brought John a hamburger.

We want to know who brought John a menu, not what John was brought. We might reasonably expect response A2 (and, in fact, response A1 as well), if the question structure were slightly altered:

Q2: Did the waitress bring John a menu?

(3) Context.

Information specific to a question´s context can help resolve focus and create response expectations. Consider the following two contexts and the affect they have on the same question:

> John had waited months for his new car. He drove it constantly. Since the weather was nice, he decided to drive to Boston.

Q3: Why did John drive to Boston?

In this context, we interpret this question to be about driving or using the new car. Answers like "He was anxious to use his car" are appropriate.

> John had loved the circus since he was a child. He was excited when he heard it was in town. Since the weather was nice, he decided to drive to Boston.

Q3: Why did John drive to Boston?

Now we are no longer as concerned with the mode of transportation or driving.

The focus here is on the destination and purpose of the trip. An answer like "He knew the circus was in town" is reasonable in this context.

This contextual notion of focus is especially problematic for modular Q/A systems, since the correct assignment of focus cannot be established without accessing memory. The focus of the question is effectively dependent on available answers in memory. Integrated Q/A systems are uniquely qualified to handle contextual focus, since they are often capable of locating answers to questions before the entire question has even been read (Dyer, 1983; Riesbeck & Martin, 1986).

(4) World knowledge.

Certain assumptions and inferences about general world knowledge can be used to resolve focus:

Q4: Why did John hang-glide to McDonalds?

A5: Because he was hungry.

A6: Because his car was broken.

We are more likely to expect an answer that addresses the strange mode of transportation (A6) than a motivation for going to McDonalds. Basic assumptions about normal transportation modes lead us to inquire about the reason for hang-gliding. If the transportation were by car, we would more reasonably expect to produce an answer like A5.

2.3 THEORY

In order to make use of our response categories and the various types of focus, we need to make some generalizations about how to view the question/answer pair. Intuitively, we might view a Y/N question as a declarative proposition that we want to confirm or deny. The question as a proposition in declarative form can be treated as a complete conceptual package. There is no missing information. We only want to confirm or deny a hypothesis. We can always perform a transformation on the question to derive the appropriate declarative form:

Did Soleil work for Hackinc?

becomes

Soleil worked for Hackinc.

If the question is viewed as a proposition, then the answer must be interpreted with respect to that proposition in order to ascertain a confirmation or denial.

This same idea is also applicable to the class of Wh-questions. However, we must slightly modify the concept. In Wh-questions we cannot produce a complete proposition and attempt to establish confirmation. Here we only want to complete a conceptual unit that has already been confirmed. For example:

Did John go to the store for milk?

As a Y/N question, we must try to confirm the proposition:

John went to the store for milk.

In a Wh-question, we already have a proposition established as a presupposition. When we ask:

Why did John go to the store?

or:

Where did John get the milk?

we know that John went to the store or that John got some milk. We seek only to complete a conceptual unit. This will generally be the format of Wh-questions.

Using our categories of question responses and focus resolution, we will develop a computational method for interpreting answers to questions based on predictive natural language understanding.

3. TECHNICAL OVERVIEW

3.1 INTRODUCTION

At this point, we can discuss the technical components of our model. The The first such component addresses the issue of knowledge representation. We will utilize a representation based on frames. Frames are abstract data structures that depict a "chunk" of information; be it an object, an event, or a concept. Given this representational format, we will take advantage of the capabilities of the natural language processor PLUM. PLUM (the Predictive Language Understanding Mechanism) is a tool or control structure for sentential language analysis (Lehnert & Rosenberg, 1985). PLUM makes no commitments to any linguistic or conceptual modes of parsing. It was not designed with a specific task in mind. Rather, it provides parsing capabilities that are flexible enough to be adapted to many needs, including question/answer understanding as described here (Lehnert, et. al., 1985).

3.2 FRAMES AND MEMORY REPRESENTATION

The memory representation utilized by PLUM is frame-based. Frames are data-structures that can represent generalized or stereotypical information about situations, events, objects, and complex concepts (Minsky, 1975). The design of specific frames used by PLUM is left to the system designer. PLUM provides only the mechanisms to create and maintain these structures. When PLUM analyzes a sentence, the verb of the sentence is frequently responsible for activating the top-level frame that structures the meaning of the sentence. Frames also have slot-value pairs that hold information about this particular instantiation of the frame. For example, the actor, place, and time that an event took place could all be organized under slots inside a conceptual frame. As an example, we will consider a frame-based representational system known as **Conceptual Dependency** (Schank, 1975; Schank & Riesbeck, 1981). This system decomposes the meaning of a sentence into a small set of primitive actions. Sentences with the same *meaning*, although lexically different, will be represented by the same conceptual dependency frame. One primitive in this representational system is a PTRANS. This refers to any transfer of physical location. A PTRANS requires an actor, object, origin, and destination as its slots. For example, the sentence:

John ran from school to town.

would be represented by the PTRANS frame:

```
PTRANS
  ACTOR       = John
  OBJECT      = John
  ORIGIN      = school
  DESTINATION = town
```

In this case, the verb *ran* triggers the use of the PTRANS primitive action. A different verb (such as *gave*) would necessitate the use of a different primitive action.

For our purposes, a question will be represented by a single frame. As we indicated earlier, we can consider a Y/N question as a complete conceptual unit that requires verification. There is no missing information. The frame for a question will be based on the action or verb of the declarative form of the question. For example:

Does John own a house?

John owns a house.

POSS-BY

AGENT = John

OBJECT = house

The complete concept is represented by the frame. In Wh-questions, information is often incomplete. Specifically, we will find an unfilled slot in the frame representing a Wh-question. The answer to the question should fill that empty slot. For example, using a PTRANS frame as described earlier:

Who went from school to town?

PTRANS

ACTOR = *?*

OBJECT = *?*

ORIGIN = school

DESTINATION = town

An acceptable answer will fill the missing slot values: *John* went from school to town.

3.3 PLUM AND PREDICTIVE PARSING

PLUM attempts to build frame-based memory structures as it moves through a sentence. Every memory structure that PLUM constructs must be defined prior to the parse by the system designer. This is accomplished by using a conceptual definition or frame template called a *prediction prototype*. These prototypes are triggered when a prediction is made about that structure. The prediction can come from a lexical item or from other prototypes. Each structure specifies a fixed set of required slots which must then be filled before that structure is instantiated. Only when a structure is instantiated can it be added to working memory.

The prediction prototype is a declarative data structure that creates slot-filling demons when triggered. These demons are processes that search for information in order to fill slots. When such information is recognized, it is placed in the slot. The prototype knows what slots are present in the frame, where to search for the appropriate slot-fillers, what form the slot-filler must take, and what slots must be filled before the structure can be instantiated and added to memory. These search and type constraints effectively describe *expectations*. The activated prediction prototype expects specific kinds of information to appear in certain locations during the course of the parse.

3.4 PROCESS MODEL

Now we can begin to detail our computational model for understanding answers to questions. Given our response categories, how might a predictive sentence analyzer prepare itself to process responses to a question? We should first consider the context in which the processing takes place. Two contextual modes are utilized. The first entails analyzing the answer without knowledge of the question. This is called **null-context** parsing. In this mode, the answer is analyzed as a conceptual unit independent of the question asked. Null-context parsing would lead to some problems if this were the sole means of interpretation available to us. Consider the following example:

A1: He has a car.

If this answer is parsed in a null-context, we cannot make any assumptions about confirmation or denial of the question proposition. The question could be:

Q1: Does John have a car?

or

Q2: Does John have a truck?

Q1 is answered affirmatively by A1 (echo response) while Q2 falls into the inconclusive response category when answered by A1. A follow-up question would then be necessary. However, we would like the natural language understanding mechanism to make confirmation decisions. This leads us to explore a second method of answer interpretation, that we refer to as **informed-context** parsing. In this context, the answer is parsed with knowledge of the question. We can then use expectations and focus indications from the question structure as the answer is interpreted. While this mode of analysis is certainly preferred, we will not abandon the null-context interpretation. While we can make many predictions with the question knowledge, it is not possible to develop expectations for *everything*. As an example, consider the response categories of gratuitous and emphatic answers. At best, we can only expect additional information related in some way to the question. We cannot be any more specific without severely limiting the robustness of the parser. Therefore, we will utilize the null-context parse as a means of creating general conceptual structures for information we cannot reasonably create predictions for. In this regard, the null-context parse acts as a general back-up mechanism that steps in when the informed-context parse is not sufficient.

Given the necessity of informed-context parsing, some representation of the question must be available to the parser. In COUNSELOR, other components of the system are responsible for the actual presentation of the question to the user. The question, at some point in the processing, was already reduced to a conceptual, frame-based structure before the generator produced text. This internal meaning representation will will serve as the basis representation for our model.

We can view a yes/no question as a declarative proposition that must be confirmed. Using the conceptual knowledge representation described earlier, the proposition would take the form of an instantiated frame which we shall refer to as the **prediction context**. For example, the question:

Does John work for ABC?

can be represented as the prediction context frame:

```
R-EMPLOYMENT
  EMPLOYEE = John
  EMPLOYER = ABC
```

Then, the process of understanding the answer would involve confirming or denying the presence of this question frame. However, if the process of understanding the answer is truly "predictive," knowledge of the question should enable us to develop expectations for the answer. These predictions, when developed in conjunction with a focus indicator, will allow the parser to correctly interpret a question's response. What constitutes a reasonable expectation for answers given the prediction context of a question? The processing requirements for the response categories varies. Consider what must happen in each of these previously discussed categories:

1. SIMPLE RESPONSE [category 1]. The keyword (yes, no) spurs a default action that automatically confirms or denies the prediction context. Pronouns, if any, should be trivially resolved from the question frame and the indication of focus.

2. GRATUITOUS INFORMATION / EMPHATIC RESPONSE [categories 2 & 6]. The keyword preface again triggers the default action that passes on the prediction context. Anything that follows (regardless of category) must have an additional memory structure. This conceptual structure will be developed in the null-context and passed along with the prediction context frame.

3. ECHO RESPONSE [category 3]. The answer should be intrinsically similar to the declarative form of the question with the possible exception of a pronoun reference and verb negation. Verb negation indicates denial of the proposition.

4. QUALIFIED RESPONSE / INCONCLUSIVE RESPONSE [categories 4 & 5]. These categories, as we might expect, create the most difficulties. In fact, as we will see in the following section, some question-answer pairs that fall into this category will have to be resolved by conceptual processing beyond the scope of this model. At best, memory structures can be created for the answer as it is presented, and then these structures are resolved against the prediction context.

To accomodate these needs, we will use **two independent, distinct modes of parsing**. These processes will operate at the same time as the answer is analyzed. The first process is the usual predictive parse of the answer into conceptual components; the null-context analysis. The other is a parse based on prediction prototypes constructed from the prediction context and the focus indicator; informed-context parsing. Keywords (yes, no) would trigger a default action that passes the prediction context frame to memory. In this case, no extra information needs to be processed. The prediction context, either confirmed or denied (depending upon the keyword) is all we need. Any additional information following the keyword (emphatic or gratuitous) will be parsed in the null-context where appropriate frame representations will be created in a bottom-up manner. These resulting representations will be passed along with the prediction context. If the response takes the form of an echoed question, the answer should parse nicely into the prediction context frame. Answers in the remaining categories (qualified and inconclusive) will only be resolved by comparing the two sets of frames to resolve any contradictions. This process will be described in more detail in the following section. For now, consider a detailed example.

As an example of the process, we will use the previous example of a Y/N question:

Does John work for ABC?

The prediction context frame for the declarative form of the question is represented by:

```
    R-EMPLOYMENT
      EMPLOYEE = John
      EMPLOYER = ABC
```

Based on this prediction context, we must construct a **constrained prototype** based on our expectations for the answer:

```
  FRAME:  R-EMPLOYMENT
    EMPLOYER   :  Features :  human
               :  Name :  'John'
    EMPLOYER   :  Features :  corporation
               :  Name :  'ABC'

    FOCUS      :  R-EMPLOYMENT
```

For each of the categories the processing will progress as follows:

1. SIMPLE RESPONSE. An answer such as "Yes." or "Yes, he did." will result in confirmation of the prediction context frame.

2. GRATUITOUS INFORMATION. If we have an answer of the form "Yes, and so does Bill." we must have two structures to represent the response. The first is the confirmed prediction context. The second is a representation of the additional information:

R-EMPLOYMENT

EMPLOYEE = Bill

EMPLOYER = ABC

3. ECHO RESPONSE. Answers of this category will be interpreted correctly by the prediction context. "He worked for ABC." or "He worked for them." will be predicted by the constrained prototype. Allowing for possible pronoun resolution, we have a frame identical to the prediction context frame.

4. QUALIFIED or INCONCLUSIVE RESPONSE. These categories create the most difficulties. Answers of this type will not, in most instances, be correctly predicted by the constrained prototype. For example, "He has done consulting for them." or "He didn't last year." do not completely meet any expectations. All we can do is create memory representations from the null-context analysis and let higher level conceptual processes resolve these representations against the prediction context.

5. EMPHATIC RESPONSE. As in the category for gratuitous information, the prediction context frame will be confirmed or denied (depending upon the keyword) and a null-context representation will be developed to handle any additional information. With the answer "Yes, he owns the corporation." we construct the frame:

POSS-BY

ACTOR = John

OBJECT = ABC

More examples and a detailed description of the answer understanding process will appear in the next section.

4. OUR MODEL

4.1 THE ALGORITHM

We begin by formalizing the algorithm for interpreting answers. The process is illustrated in Figure 1.

1. The parser receives the prediction context frame and focus indication from memory.

2. Using code external to PLUM, the *expectation module* builds the constrained prototype based on the prediction context frame.

3. The system begins processing the user response. It is analyzed with both the informed-context and null-context parsing mechanisms.

4. At [1] we continue processing based on the presence of a keyword in the input (prefaced or open class).

5. If the answer is in the prefaced class, the keyword indicates what form the prediction context takes when it is passed on to memory [2]. If the keyword confirms, then the prediction context as it stands is passed on; otherwise it is flagged false in the case of denial.

6. Any information that follows [4] (in either confirmation or denial) is considered emphatic or gratuitous and is passed along with the question prediction frame. This data is interpreted with the null-context conceptual information processing mechanisms.

7. If no keyword is present (open class), the system will attempt to parse the input using informed and null-context conceptual mechanisms.

8. If the constrained prototype (informed context) is not instantiated [3], the parser can attempt to resolve the null-context structures with the partially instantiated question constrained prototype. In the cases where the answer cannot be determined by the parser, all frames, partial and instantiated, will be passed on to other components of the system for conceptual resolution.

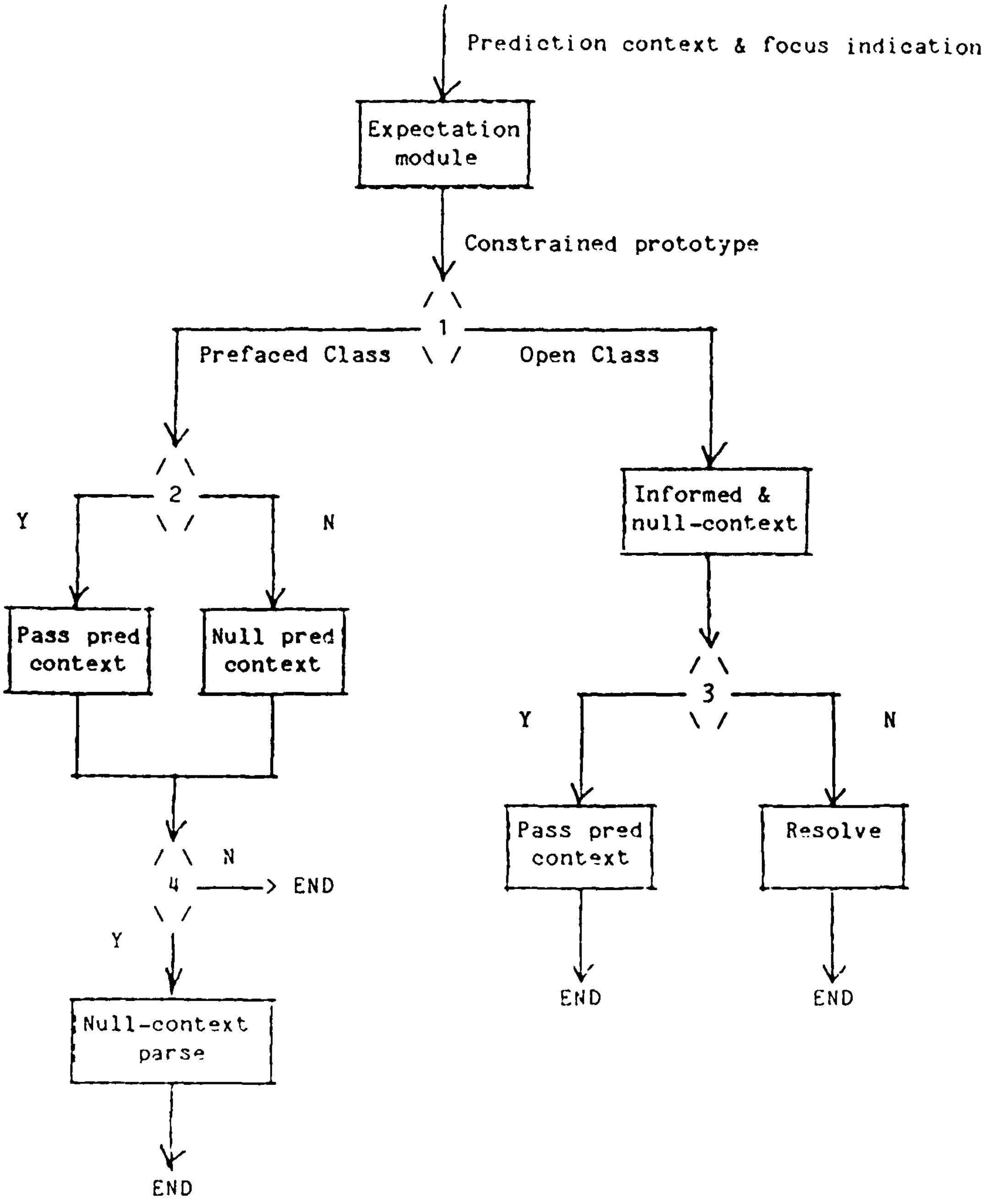

Prediction context & focus indication
Expectation module
Constrained prototype
1
Prefaced Class
Open Class
2
Y
N
Pass pred context
Null pred context
Informed & null-context
3
Y
N
Pass pred context
Resolve
END
END
4
N
END
Y
Null-context parse
END

Figure 1

4.2 AN EXAMPLE

Consider the following predicate that must be confirmed by the system:

R-EMPLOYMENT (John , ABC)

In the declarative form, we will consider the frame:

R-EMPLOYMENT

EMPLOYEE = John

EMPLOYER = ABC

The importance of focus indication becomes apparent when we consider some possible question structures based on the same predicate. The focus for this seemingly innocent proposition can be placed on either the employment relation, employee, or employer.

1. Did John work for ABC?

   Focus is not clearly placed on any item. However, if one must be considered, it is probably the employee. The question structure implies this by by placing the employee in the subject spot of the sentence. If the implied question hypothesis is incorrect, we would expect the correction to involve the employee. For example, it is more reasonable to expect an answer like "No, he worked for XYZ." than one correcting the employer, "No, Howard did."

2. Did ABC employ John?

   This is similar to the previous example except that we now expect correction to be in terms of the employer. For example, "No, they employed Howard." is now a reasonable response while "No, XYZ did." is not.

3. ABC employed John, didn´t they?

   In this case, the focus is stronger on the employer. We want to know who employed John if not ABC.

4. John worked for ABC, didn´t he?

   The focus now falls on the employee. The question implies a need to know who employed him.

5. Was it John who worked for ABC?

   The focus in this case is clearly on the employee, John. It would seem very unnatural to have an answer with a reference to the employer.

6. Was it ABC who employed John?

   Similarly, the focus is clearly on the employer.

Assuming the focus is on the work-for relation, we might develop the constrained prototype as we indicated earlier:

```
FRAME:  R-EMPLOYMENT

  ACTOR          : FEATURES:  human

                 : NAME     :  ´John´

  EMPLOYER       : FEATURES:  corporation

                 : NAME     :  ´ABC´
```

As an example, we will consider a response with gratuitous information to demonstrate what occurs at each stage of the processing on a word by word basis. Comments in the program trace are italicized.

```
******************** PLUM parse beginning ********************

Input:  "No, he worked for XYZ."


Processing a new word: NO

            ____________________________

** keyword NO recognized

R-EMPLOYMENT1-nil

            employee = JOHN

            employer = ABC
```

The keyword triggers the default mechanism that passes the prediction context frame. The frame must be marked to indicate a negation or denial of the question.

```
Processing a new word:  HE.

            ____________________________

Activating a new memory structure ...

NOUN-PHRASE1

Searching for headnoun in last word

      Search succeeds

Instantiating a new memory structure ...

NOUN-PHRASE1:

            headnoun = HE

            determiner = nil
```

```
          modifier = nil
    ** resolution concept added  NOUN-PHRASE1 (human1 name John)
```

Since we have additional information we continue with a null-context parse. The constrained prototype and its subsequent question expectations are no longer utilized since the prediction context has been denied. However we must develop the null-context conceptual structures for the extra data. The pronoun "he", given the context, is resolved to "John."

```
Processing new word:  WORKED.
```

```
Activating a new memory structure ...
R-EMPLOYMENT2
Searching for employee in last hum-referent
      Search succeeds
Searching for employer in next corp-referent
```

The verb "worked" activates an EMPLOYMENT frame with slots for the EMPLOYER and EMPLOYEE. The previous pronoun "he" (which has been resolved to John) fills the EMPLOYEE slot. A demon is created to search for the employer slot-filler in an appropriate future referent.

```
Processing a new word:  FOR.
```

```
Activating a new memory structure ...

PREP-PHRASE1

Searching for prep in last word

      Search succeeds

Searching for object in next referent
```

The word "for" creates an expectation for a prepositional phrase.

```
Processing a new word:  XYZ.
```

```
Activating a new memory structure ...

NOUN-PHRASE2

Searching for headnoun in last word

      Search succeeds

Instantiating a new memory structure ...

NOUN-PHRASE2

            headnoun = XYZ

            determiner = nil

            modifier = nil

** resolution concept added  NOUN-PHRASE2 (corp1 name XYZ)

      Search succeeds
```

Prepositional frame completed.

```
Instantiating a new memory structure ...
```

```
PREP-PHRASE1
          prep = for
          object = corp1
                  name = XYZ
Search succeeds
```

This is also the expected referent for the conceptual employment frame.

```
Instantiating a new memory structure ...
R-EMPLOYMENT2
          employer = human1
                  name = John
          employer = corp1
                  name = XYZ
```

The final component to fill the null-context conceptual employment frame is recognized.

```
************************* END PARSE ************************
```

With the input ended, we are now left with two conceptual frames to represent the user answer to the question:

```
        R-EMPLOYMENT - * flagged false
          EMPLOYEE = John                         INFORMED-CONTEXT
          EMPLOYER = ABC
```

R-EMPLOYMENT

EMPLOYEE = John NULL-CONTEXT

EMPLOYER = XYZ

Notice that it is not enough to pass only the second frame indicating John's actual employment. Without the first negated frame, we cannot assume that he does *not* work for ABC. It is always possible for people to hold two jobs.

4.3 PROBLEMS

It is appropriate at this point to indicate areas of the model that are in need of further research. Although we have built a strong foundation with the initial system development, there are complex issues yet to be resolved.

To begin with, the mechanism to perform resolution between a partially instantiated constrained prototype and the null-context conceptual structures could be quite complex. The reasons for inadequate question predictions can be diverse and this "pattern matcher" that will be necessary will have to account for as many alternatives as possible without decreasing confidence in its ability to correctly resolve question/answer pairs.

Further, the level of knowledge and inference developed in the parser can have an impact on the robustness of the system. Consider the following example:

Q: Did Malcolm enter into a noncompetition agreement with ABC, Inc.?

A: He signed it in 1980.

Clearly, the answer is "yes." With a keyword preface, the case is trivial. However, without the keyword, the parser must understand that entering into an agreement is the same as signing "it" (which would be resolved to the noncompetition agreement). The could require rather complex inferencing mechanisms that may be beyond the scope of the parser. This is dependent upon the use and separation of knowledge and expertise in the given system.

5. CONCLUSIONS

Intuitively, it seems that anyone asking a question is going to have very strong predictions about likely answers that might be returned. Even so, the problem of understanding answers to questions is one application for predictive language processing that remains relatively untouched by computational models for natural language comprehension. This oversight is most likely due to the fact that computers are usually cast in the role of the question answerer rather that the question asker.

In trying to build a question-asking facility for the COUNSELOR project, we were forced to confront the issue of predictive language processing when a system tries to understand answers to questions. Some answers conform nicely to likely expectations and others do not. This makes it necessary to create a facility that can take advantage of predictions when appropriate, while remaining capable of general natural language processing in those cases where all predictions fail. The problem is one of balancing a strongly predictive top-down processor against a general bottom-up mechanism in a way that allows for graceful interactions between the two language processing strategies.

Our solution to the problem has been to maintain two separate processes for sentence analysis which theoretically operate in parallel with one another. At the end of the sentence we check the the predictive (informed-context) analysis for a quick confirmation or refutation of the question asked. The results of the more general bottom-up (null-context) analysis are then consulted for any additional information or perhaps information that was completely unexpected.

While it would be more elegant to collapse both facilities into a single mechanism, it is not altogether unreasonable to posit two separate modes of analysis. If this approach seems psychologically improbable, it is only because we have not addressed the question of exactly how these two mechanisms might interact with one another during sentence analysis. In a psychologically plausible model, we could assume some amount of short term memory that allows us to save a small number of lexical items as we move through the sentence. This short term memory buffer would give us some allowance for error if we needed to abandon a predictive analysis and effectively start over again. People seem to do this. In casual conversation one is sometimes aware of such revisions: "Oh, I thought you were going to say (something else)," signals a failed expectation of this type.

Unfortunately, we cannot expect to justify this (or any other model) of question asking and answer understanding on the basis of subjective anecdotes. In the absence of experimental data on how people understand answers to questions, we can only hypothesize possible computational models and address the consequences of such a model within computer simulations. Our work in the COUNSELOR project represents a modest first step in this regard.

6. REFERENCES

Ashley, K., & Rissland, E. "Toward Modelling Legal Argument." *CPTM 7*, Department of Computer & Information Science, University of Massachusetts, Amherst, MA, May 1985.

Bates, M. The Theory and Practice of Augmented Transition Network Grammars. In L. Bolc (Ed.), *Natural Language Communication with Computers*. Berlin: Springer Verlag, 1978.

Charniak, B. "A Parser with Something for Everyone." *Technical Report CS-70*, Department of Computer Science, Brown University, Providence, RI, 1981.

Davies, D.J.M., & Isard, S.D. Utterances as Programs. In B. Meltzer & D. Michie (Eds.), *Machine Intelligence 7*. John Wiley & Sons, 1972.

DeJong, G. Prediction and Substantiation: A New Approach to Natural Language Processing. *Cognitive Science*, 1979, 3, 251-273.

Dyer, M. *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension*. The MIT Press, 1983.

Graesser, A. & Black, J. (Eds.) *The Psychology of Questions*, Lawrence Erlbaum Associates Publishers, 1985.

Grosz, B. Focus Spaces: A Representation of the Focus of Attention of a Dialog. In Donald Walker (Ed.), *Understanding Spoken Language*. North-Holland, 1978.

Grosz, B. Focusing and Description in Natural Language Dialogues. In A. Joshi, B.L. Webber, & Sag (Eds.), *Elements of Discourse Understanding*. Cambridge University Press, 1981.

Lehnert, W. *The Process of Question Answering*. New Jersey: Lawrence Erlbaum Associates Publishers, 1978.

Lehnert, W. & Rosenberg, S. "The PLUM Users Manual." *CPTM 1*, Department of Computer & Information Science, University of Massachusetts, Amherst, MA, May 1985.

Lehnert, W., Narasimhan, N., Draper, B., Stucky, B., & Sullivan, M. "Experiments with PLUM." *CPTM 2*, Department of Computer & Information Science, University of Massachusetts, Amherst, MA, May 1985.

Marcus, M.P. *A Theory of Syntactic Recognition for Natural Language*. Cambridge, MA: MIT Press, 1980.

McDonald, D. & Pustejovsky, J. "TAGs as a Grammatical Formalism for Generation." *CPTM 5*, Department of Computer & Information Science, University of Massachusetts, Amherst, MA, June 1985.

Minsky, M. A Framework for Representing Knowledge. In P.H. Winston (Ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill Book Company, 1975.

Riesbeck, C. & Martin, C. "Direct Memory Access Parsing." *Cognitive Science*, (in press), 1986.

Riesbeck, C. & Schank, R. "Expectation-Based Analysis of Sentences in Context." *Research Report 78*, Department of Computer Science, Yale University, New Haven, CT, 1976.

Schank, R.C. *Conceptual Information Processing*. Amsterdam: North-Holland, 1975.

Schank, R.C. & Riesbeck, C. *Inside Computer Understanding: Five Programs Plus Miniatures*. New Jersey: Lawrence Erlbaum Associates, 1981.

Sidner, C. Focusing in the Comprehension of Definite Anaphora. In M. Brady & R. Berwick (Eds.), *Computational Models of Discourse*. The MIT Press, 1983.

Waltz, D.L. & Pollack, D.B. "Phenomenologically Plausible Parsing." *Proceedings of the 1984 American Association for Artificial Intelligence Conference*, pp. 335-339.

Woods, W.A. "Transition Network Grammars for Natural Language Analysis." *Communications of the ACM*, 13(10), 1970.

END

1-87

DTIC